Supplementary Material of Open-world Text-specified Object Counting

Niki Amini-Naieni[1]
niki.amini-naieni@eng.ox.ac.uk
Kiana Amini-Naieni[2]
kamininaieni@ucdavis.edu
Tengda Han[1]
htd@robots.ox.ac.uk
Andrew Zisserman[1]
az@robots.ox.ac.uk

[1] Visual Geometry Group (VGG), University of Oxford, UK
[2] University of California, Davis, USA

Contents

Section 1 describes additional implementation details about the CounTX training algorithm. Section 2 presents and analyzes CounTX's performance on additional datasets. Section 3 compares the quality of different frozen image encoder backbones for the counting task. Further information about FSC-147-D is provided in section 4. The Python code in `model.py` is explained in section 5. Section 6 illustrates additional density maps generated by CounTX to supplement the images already included in the paper. Finally, section 7 discusses known weaknesses of CounTX to be improved on in future work.

1 Additional Training Implementation Details

In this section, additional implementation details about the CounTX augmentation scheme and the construction of the ground truth density maps are discussed. During training, images are augmented with a probability of $\frac{2}{5}$. If augmentation is applied, either the augmentation pipeline presented in Table 1 is used with a probability of $\frac{3}{8}$, or a scalable mosaicking scheme is employed with a probability of $\frac{5}{8}$. For the scalable mosaicking scheme, if an image contains greater than or equal to seventy objects to be counted, the same image is cropped four times to create the mosaicked image. Otherwise, four different training images are used. Cropping and combining the same image means the number of objects can be increased. Cropping and combining four different images teaches the model to distinguish between different semantic categories using the text descriptions. α-channel blending is applied to soften the sharp borders between the four different crops in the mosaicked image. These augmentation techniques were adopted from CounTR [7], which can be referenced for further details. To construct the ground truth density maps, the provided annotations with ones at the centers of the objects to be counted and zeros elsewhere were filtered with a Gaussian kernel with x and y standard deviations of one and a radius of four.

Augmentation	**Settings**
Gaussian Noise	mean: 0 standard deviation: 0.1
Color Jitter	brightness factor: 0.25 contrast factor: 0.15 saturation factor: 0.15 hue factor: 0.15
Gaussian Blur	kernel size: (7, 9) standard deviation: sampled uniformly from $[0.1, 2]$
Random Affine	rotation: sampled uniformly from $[-15°, 15°]$ scale factor: sampled uniformly from $[0.8, 1.2]$ translation factor (x, y): sampled uniformly from $[-0.2, 0.2] \times [-0.2, 0.2]$ shear (x, y): sampled uniformly from $[-10°, 10°] \times [-10°, 10°]$
Horizontal Flip	horizontally flipped with probability $\frac{1}{2}$

Table 1: CounTX augmentation pipeline. The augmentations are applied during training with a probability of $\frac{3}{20}$ in the top-to-bottom order of the rows in the table.

Method	Method Type	How to Specify the Class	Val-COCO MAE	Val-COCO RMSE	Test-COCO MAE	Test-COCO RMSE
RetinaNet [6]	Closed-set	Text (class name)	63.57	174.36	52.67	85.86
Faster-RCNN [13]	Closed-set	Text (class name)	52.79	172.46	36.20	79.59
Mask-RCNN [7]	Closed-set	Text (class name)	52.51	172.21	35.56	80.00
CounTX (FSC-147-D)	Open-set	**Text (FSC-147-D)**	**29.39**	**101.56**	**12.15**	**25.49**
FamNet [12]	Open-set	Visual Exemplars	39.82	108.13	22.76	45.92
BMNet+ [15]	Open-set	Visual Exemplars	26.55	93.63	12.38	24.76
CounTR [9]	Open-set	Visual Exemplars	24.66	83.84	10.89	31.11
LOCA [1]	Open-set	Visual Exemplars	16.86	53.22	10.73	31.31

Table 2: Performance of closed-set and open-set models on the Val-COCO and Test-COCO subsets of COCO [5] and FSC-147 [12]. Methods in the bottom four rows are grayed out because they use visual exemplars, which provide more information than class descriptions.

2 Additional Experiments on Other Datasets

2.1 Val-COCO & Test-COCO

A straightforward approach to object counting is to enumerate all the class instances produced by pre-trained object detectors such as RetinaNet [6] and Faster-RCNN [13] or by an instance segmentation model such as Mask-RCNN [7]. Therefore, it is instructive to investigate how CounTX performs compared to these models. However, unlike CounTX, RetinaNet, Faster-RCNN, and Mask-RCNN are closed-set methods and, thus, are limited to counting instances of classes they were trained on. On the other hand, CounTX is an open-set model and, as a result, can count instances of arbitrary classes.

The FSC-147 [12] dataset provides Val-COCO and Test-COCO, image subsets of the COCO [5] dataset. RetinaNet, Faster-RCNN, and Mask-RCNN have been trained to categorize objects into the classes present in these subsets. As a result, CounTX was evaluated against these methods using Val-COCO and Test-COCO. As shown in table 2, CounTX performs better than all three closed-set methods and the class-agnostic counting model FamNet. However, unlike FamNet, CounTX does not require any visual exemplars for inference.

2.2 CARPK

CounTX is evaluated quantitatively and qualitatively on the CARPK [6] dataset to demonstrate its ability to generalize to datasets other than FSC-147 [12]. The CARPK dataset for counting cars contains overhead images of parking lots captured by drone cameras. The CARPK training set includes 989 images, and the CARPK test set includes 459 images. CARPK contains photos of 90,000 cars altogether.

CounTX was trained and evaluated in multiple settings for CARPK [6] as shown in Table 3. For the fifth and sixth rows in Table 3, CounTX was trained on FSC-147 [12] and evaluated on the CARPK test set. For the seventh and eight rows in Table 3, CounTX was jointly trained on data from FSC-147 and CARPK and evaluated on the CARPK test set. Specifically, during joint training, each batch was constructed from data from either FSC-147 or CARPK. The batches were composed and shuffled randomly, and augmentation was only applied to data from FSC-147. Table 3 illustrates CounTX's performance using different potential responses to the query "what object should be counted" for CARPK, as indicated by the third column.

As shown in Table 3, CounTX performs competitively compared to closed-set counting methods on CARPK [6]. Methods with asterisks in Table 3 were trained specifically for car detection, while the other closed-set methods can count instances of other classes. With

Method	Method Type	How to Specify the Class	CARPK MAE	CARPK RMSE
Faster-RCNN [13]	Closed-set	Text (class "car")	39.88	47.67
One-look Regression* [8]	Closed-set	Text (class "car")	21.88	36.73
RetinaNet [8]	Closed-set	Text (class "car")	16.62	22.30
HLCNN* [8]	Closed-set	Text (class "car")	2.12	3.02
CounTX (FSC-147)	Open-set	Text (description "cars")	11.72	14.86
CounTX (FSC-147)	Open-set	Text (description "car")	11.64	14.85
CounTX (FSC-147 & CARPK)	Open-set	Text (description "cars")	8.89	11.42
CounTX (FSC-147 & CARPK)	Open-set	**Text (description "the cars")**	**8.13**	**10.87**
FamNet (CARPK) [12]	Open-set	Visual Exemplars	18.19	33.66
CounTR (FSC-147 & CARPK) [8]	Open-set	Visual Exemplars	5.75	7.45
SAFECount (FSC-147 & CARPK) [16]	Open-set	Visual Exemplars	5.33	7.04

Table 3: Performance of closed-set and open-set models on the CARPK [8] dataset. Methods in the bottom three rows are grayed out because they use visual exemplars, which provide more information than class descriptions. Methods with asterisks were trained specifically for car counting, while other methods can count objects of other classes as well.

and without being trained on data in CARPK, CounTX performs better than the few-shot class-agnostic counting method FamNet [12] trained on data from CARPK. It is interesting to consider whether training FamNet on CARPK damages its performance on FSC-147 [12] significantly. Similarly, the fine-tuning of models such as CounTR [7] and SAFECount [16] could cause them to lose their generality. The joint training procedure for CounTX avoids this issue by ensuring that the model performs well on both FSC-147 and CARPK during the final optimization process. The performance of CounTR and SAFECount pre-trained on FSC-147 and then fine-tuned on CARPK is shown in the bottom two rows of Table 3. Figure 1 illustrates the effectiveness and generality of CounTX trained only on data from FSC-147 and evaluated on the CARPK test set.

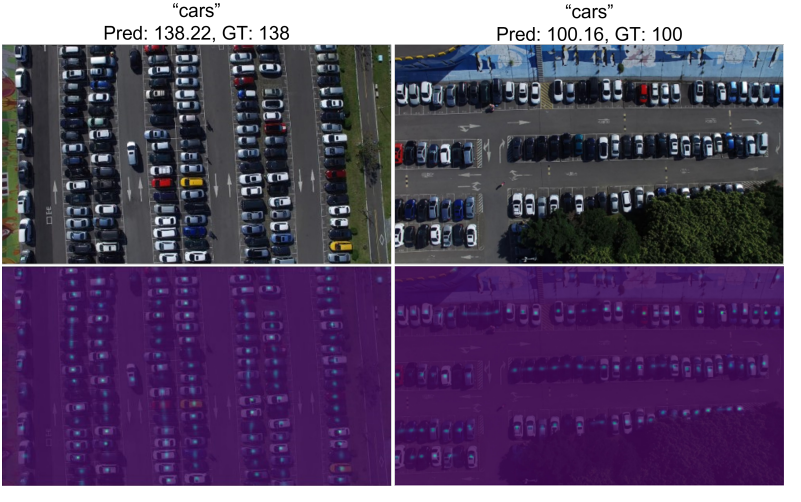

Figure 1: Density maps produced by CounTX, trained on FSC-147 [12], when applied to the CARPK [8] test set with no fine-tuning.

3 Additional Ablation Study: Image Encoder Backbones

To measure the quality of different image features for object counting, three different CLIP image encoder backbones were frozen and used to train CounTX with visual exemplars instead of text. The CounTX *text* encoder was replaced with the exemplar encoder (4 convolutional layers followed by global average pooling) from CounTR [7]. The training and inference procedures from CounTR were also adopted. As shown in Table 4, compared to the other two CLIP models, the image encoder used in the main paper for CounTX, ViT-B-16, performs competitively on FSC-147 [12].

The image encoder from CounTR [7] (pre-trained on ImageNet and then with self-supervised patch reconstruction on FSC-147 [12]) was also frozen and evaluated. It has been mentioned multiple times in the literature [9, 17] that CLIP image encoders may not provide as rich spatial features out-of-the-box as other more generally trained image backbones. This point is consistent with the results in Table 4, as the frozen CounTR image encoder, pre-trained with self-supervision, performs generally better than all three CLIP image encoders on FSC-147. However, the CounTR image encoder backbone does not have an available joint text-image embedding space as the CLIP image encoder backbones do.

Image Encoder Backbone	Pre-training Method	Embedding Dimension	Spatial Feature Map Shape	Validation		Test	
				MAE	RMSE	MAE	RMSE
CounTR [7]	ImageNet and SSL	512	24×24	15.53	53.01	14.93	94.38
RN50x16 [8]	YFCC100M Subset [13]	768	12×12	32.84	98.37	26.96	100.31
ViT-L-14-336 [8]	YFCC100M Subset [13]	768	24×24	27.35	81.73	**22.72**	96.34
ViT-B-16 [8]	**LAION-2B [14]**	**512**	$\mathbf{14 \times 14}$	**26.37**	**71.28**	24.96	**91.64**

Table 4: Performance of different frozen image encoder backbones on the FSC-147 [12] 3-shot visual exemplar counting task. The last three rows contain data from CLIP image encoder backbones, while the first row contains data from the CounTR image encoder backbone. SSL stands for self-supervised learning.

4 Details of the FSC-147-D Dataset

The file `FSC-147-D.json` contains the FSC-147-D dataset with class descriptions for the images in FSC-147 [12]. `FSC-147-D.json` is available at https://www.robots.ox.ac.uk/~vgg/research/countx/. While FSC-147 provides class names, FSC-147-D contains responses to the query "what object should be counted?" 92.4 % of the class descriptions in FSC-147-D (5668 class descriptions) are the class names in FSC-147 with "the" prepended to them. The remaining 7.6 % of the class names in FSC-147 (467 class names) required more complex rephrasing to convert them to class descriptions in FSC-147-D. Figure 2 illustrates the distribution of the number of words for the class names in FSC-147 and the distribution of the number of words for the class descriptions in FSC-147-D.

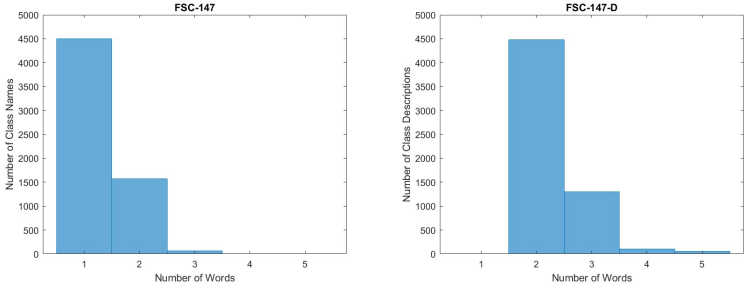

Figure 2: Histograms of the number of words in the class names for FSC-147 (left) and the number of words in the responses to the question "what object should be counted?" for FSC-147-D (right). The histograms show that the class descriptions in FSC-147-D are more prolific than the class names in FSC-147.

5 Counting Network Code

The file `model.py` contains the class `CountingNetwork` used to initialize CounTX as shown in Figure 3 below.

```
class CountingNetwork(nn.Module):
    def __init__(
        self,
        img_encoder_num_output_tokens=196,
        fim_embed_dim=512,
        fim_depth=2,
        fim_num_heads=16,
        mlp_ratio=4.0,
        norm_layer=nn.LayerNorm,
    ):
        super().__init__()
```

Figure 3: Header of the `CountingNetwork` class definition. A CounTX model is initialized in code using the `CountingNetwork` class.

The forward method of the `CountingNetwork` class shows that CounTX takes both images and text as inputs. This forward function is included in Figure 4.

```
def forward(self, imgs, counting_queries, shot_num):
    img_tokens = self.forward_img_encoder(imgs)
    txt_tokens = self.foward_txt_encoder(counting_queries, shot_num)
    fim_output_tokens = self.forward_fim(img_tokens, txt_tokens)
    pred = self.forward_decoder(fim_output_tokens)
    return pred
```

Figure 4: Forward method for a `CountingNetwork` instance. A `CountingNetwork` instance uses both images and text descriptions to infer the object count.'fim' stands for feature interaction module.

A worked example of how to use CounTX to infer the object count in an image with a text description is provided at the bottom of the `CountingNetwork` class definition in a long comment. The start of this worked example is shown in Figure 5.

```
"""
Worked Example:

# Model and image file names.
model_file_name = "./bmvc23_model.pth"
image_file_name = "./espresso_spoons.jpg"

# Load model.
model = main_counting_network()
checkpoint = torch.load(model_file_name, map_location="cpu")
model.load_state_dict(checkpoint["model"], strict=False)

# Define preprocessing.
tokenizer = open_clip.get_tokenizer("ViT-B-16")
```

Figure 5: A worked example of how to infer the object count with CounTX appears at the bottom of `model.py` in a long comment.

6 Additional Counting Image Examples

6.1 FSC-147

This section presents and comments on additional density maps produced by CounTX when applied to the FSC-147 [12] test set. Results are shown in Figures 6, 7, 8, 9, and 10.

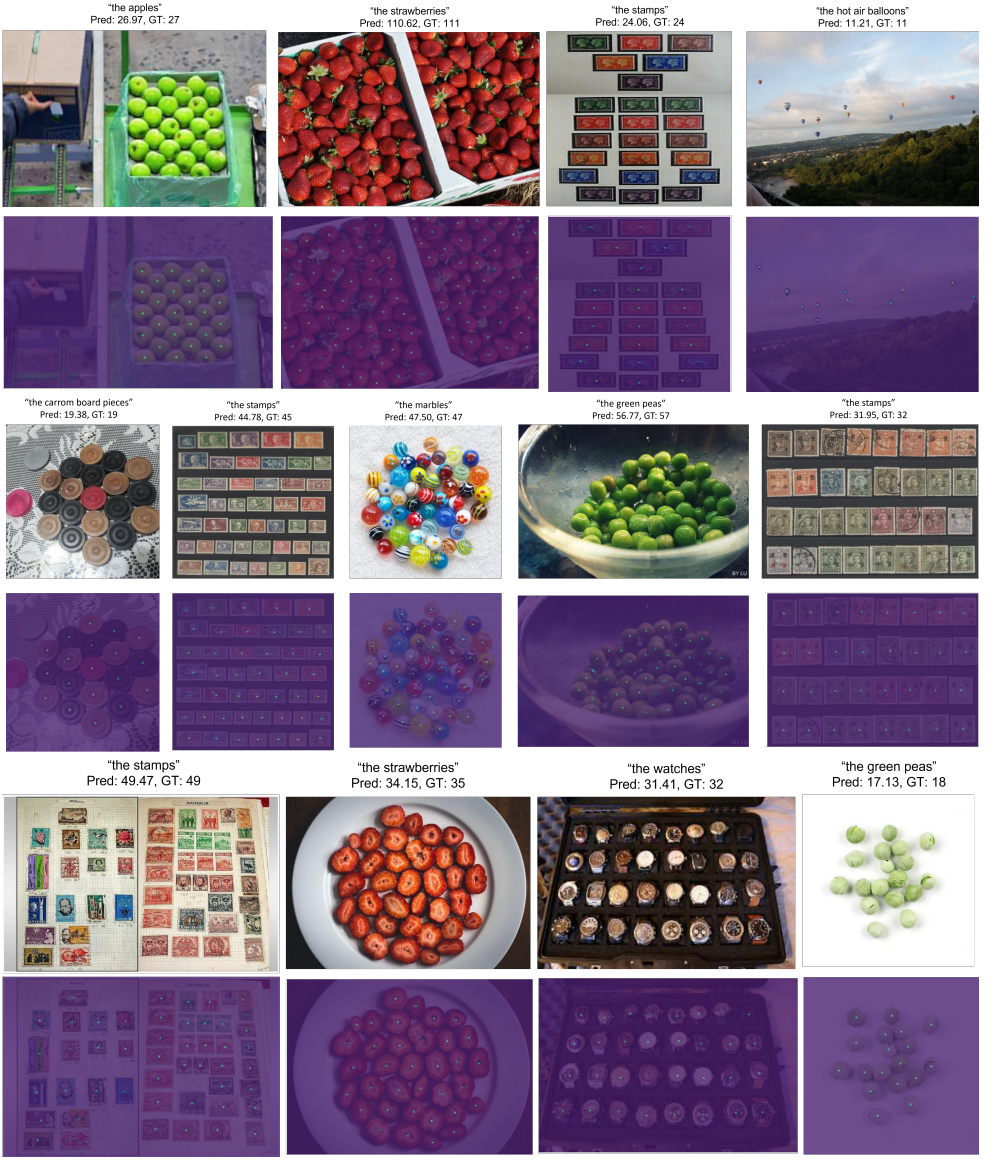

Figure 6: Density maps produced by CounTX when applied to the FSC-147 [13] test set. CounTX is able to count the hot air balloons in the rightmost image in the top row despite how small they are. CounTX also correctly counts only the carrom board pieces and excludes the extraneous circular objects in the leftmost image in the third row. Despite the variance in the color and shape of the stamps, CounTX counts them.

Figure 7: Density maps produced by CounTX when applied to the FSC-147 [12] test set. In the third image in the third row, CounTX only counts dishes with finger food in them, correctly excluding empty dishes. CounTX only counts the apples and not the leaves even though there are more leaves than apples in the leftmost image in the fifth row. CounTX includes both peas inside the soup and around the soup in the final object count when applied to the second image in the fifth row. Despite the glare on the comic books, CounTX counts all of them in the rightmost image in the third row.

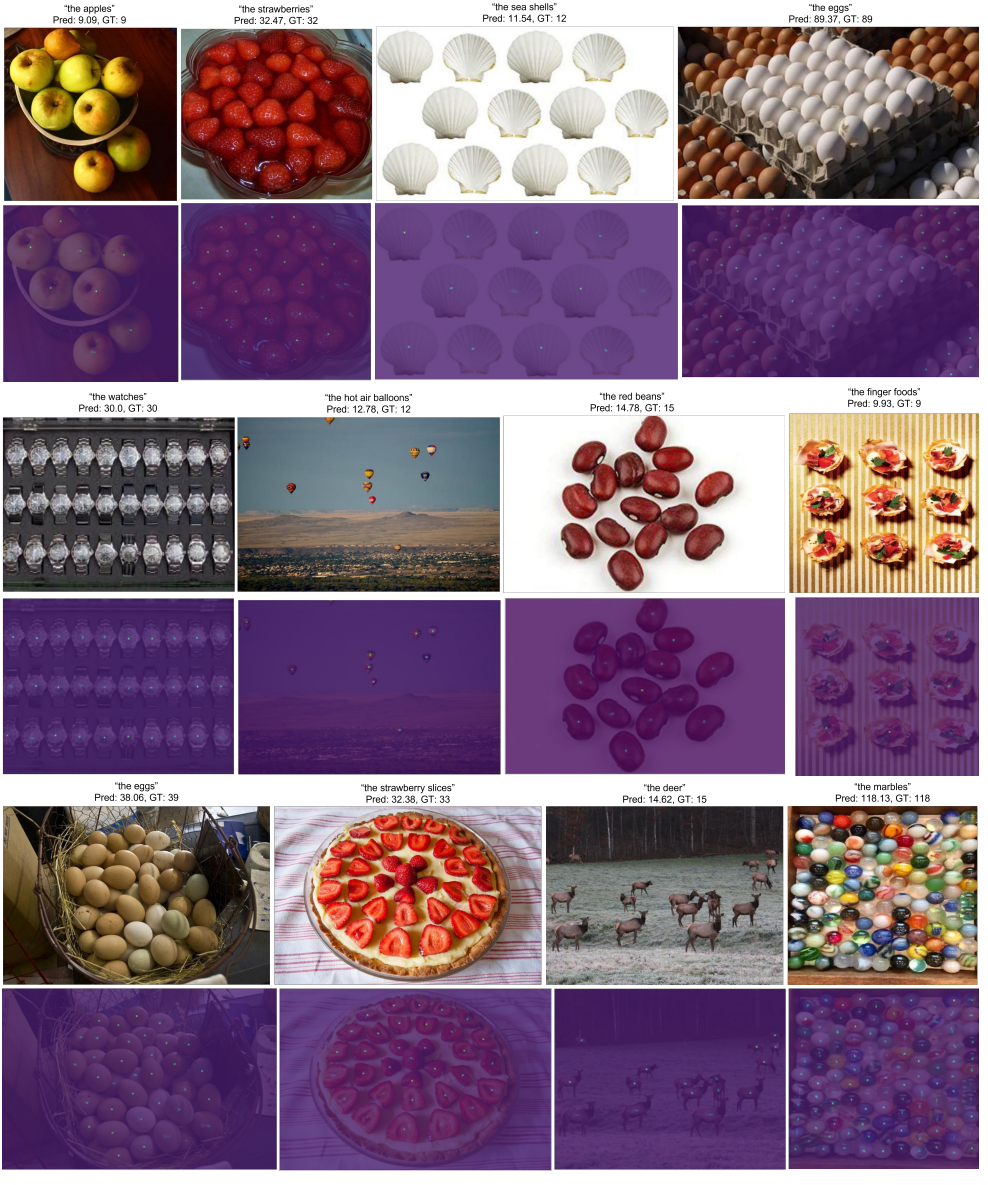

Figure 8: Density maps produced by CounTX when applied to the FSC-147 [13] test set. CounTX can count small and large numbers of objects as shown by its performance on the images of the apples and marbles above. CounTX correctly includes both the brown and white eggs in its final estimate in response to the class description "the eggs."

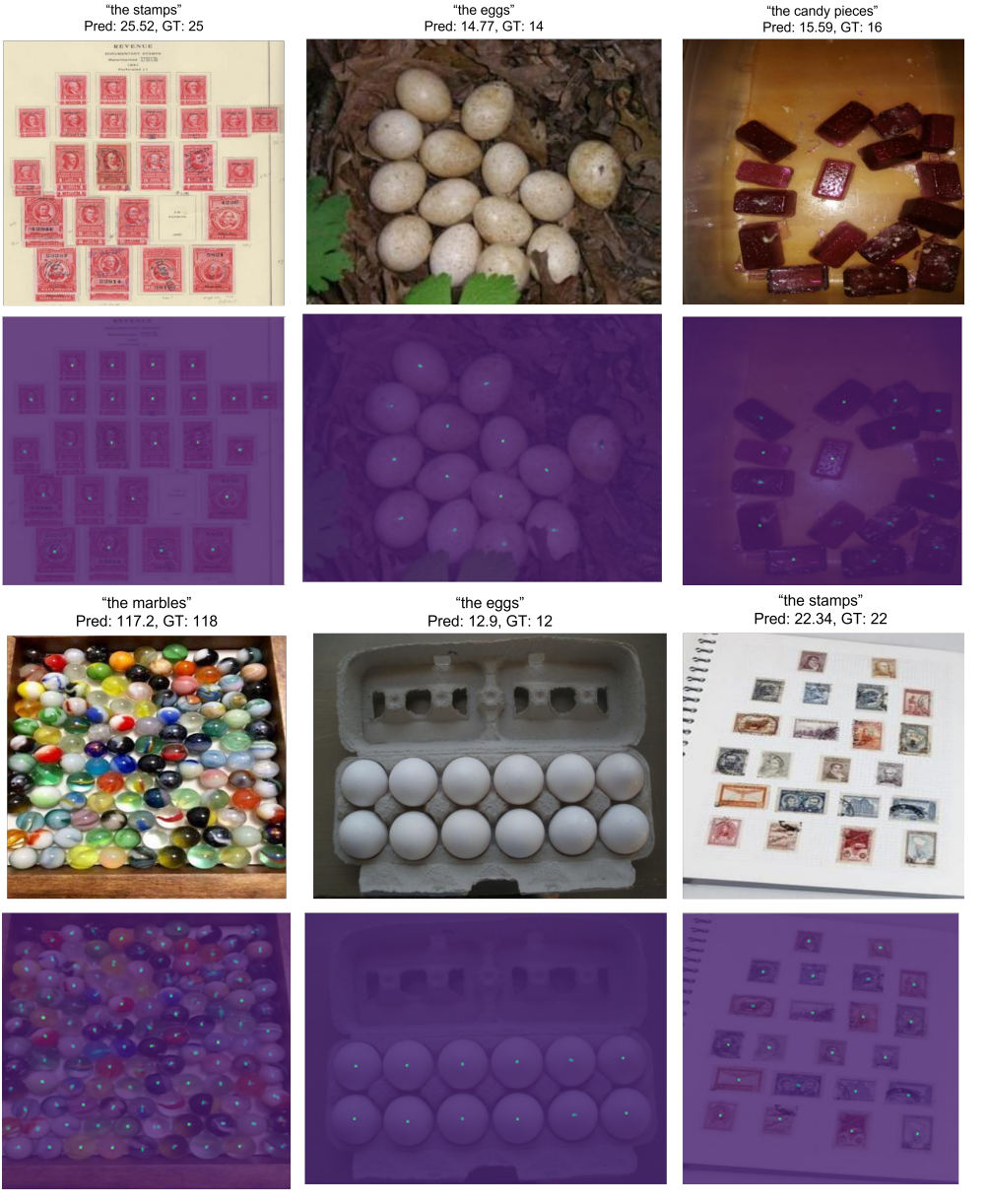

Figure 9: Density maps produced by CounTX when applied to the FSC-147 [12] test set. Notice how CounTX counts all the stamps in the rightmost image in the third row despite the change in viewpoint and the candy pieces in the rightmost image in the top row despite the glare.

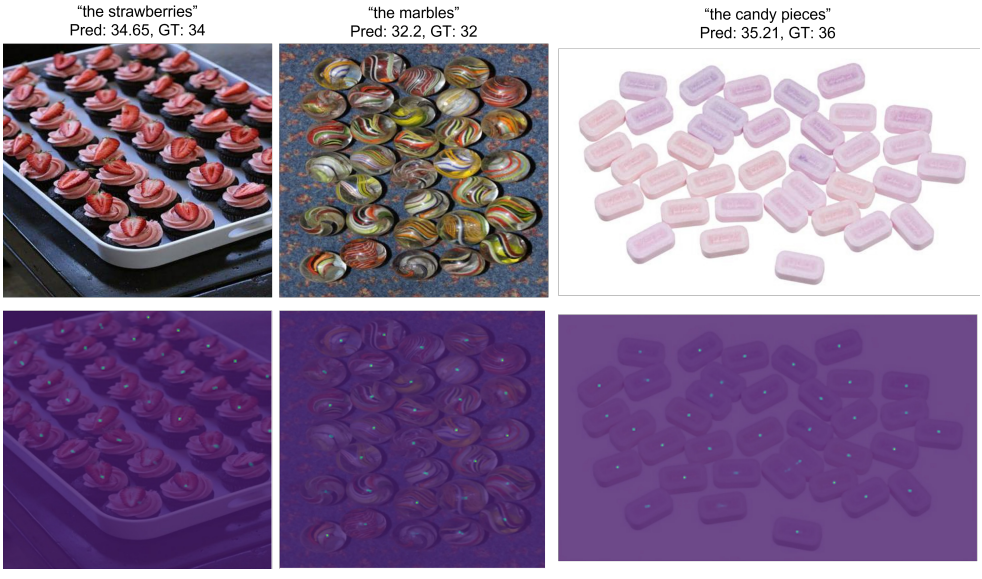

Figure 10: Density maps produced by CounTX when applied to the FSC-147 [13] test set. CounTX counts the strawberries in leftmost image in the top row even though they are placed on cupcakes as toppings.

6.2 CountBench

Text descriptions were created for a subset of CountBench [10]. These descriptions are more detailed and longer in general than the descriptions in FSC-147-D. CountBench also only contains images with at most 10 objects. On the other hand, images in FSC-147 [17] contain at minimum 7 objects and at most 3731 objects with an average of 56 objects per image. Therefore, it is interesting to investigate how CounTX performs on the CountBench subset given that CounTX has never been trained on images with under 7 objects. In this section, qualitative examples are provided and commented on from such an investigation. Results are shown in Figures 11 and 12.

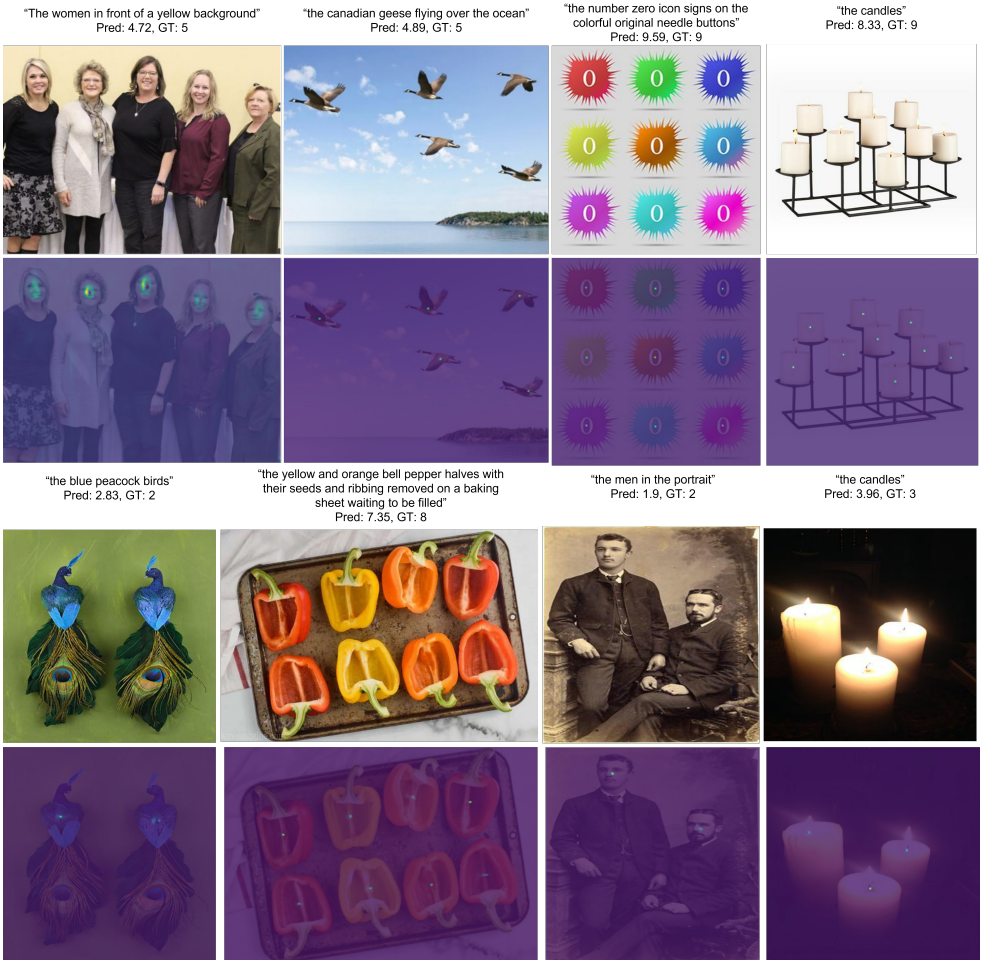

Figure 11: Density maps produced by CounTX when applied to the CountBench [10] subset. Even though CounTX was never trained to count people, it can count the women in the leftmost image in the top row and the men in the third image in the third row. CounTX estimates that there are almost exactly 2 men in the third image in the third row, even though no image in FSC-147 [17], the dataset CounTX was trained on, has under 7 objects. The class descriptions in FSC-147-D are very simple compared to the long and detailed description of the bell peppers in the third row. Despite this, CounTX provides a reasonable estimate for the count of the bell pepper halves given this long description.

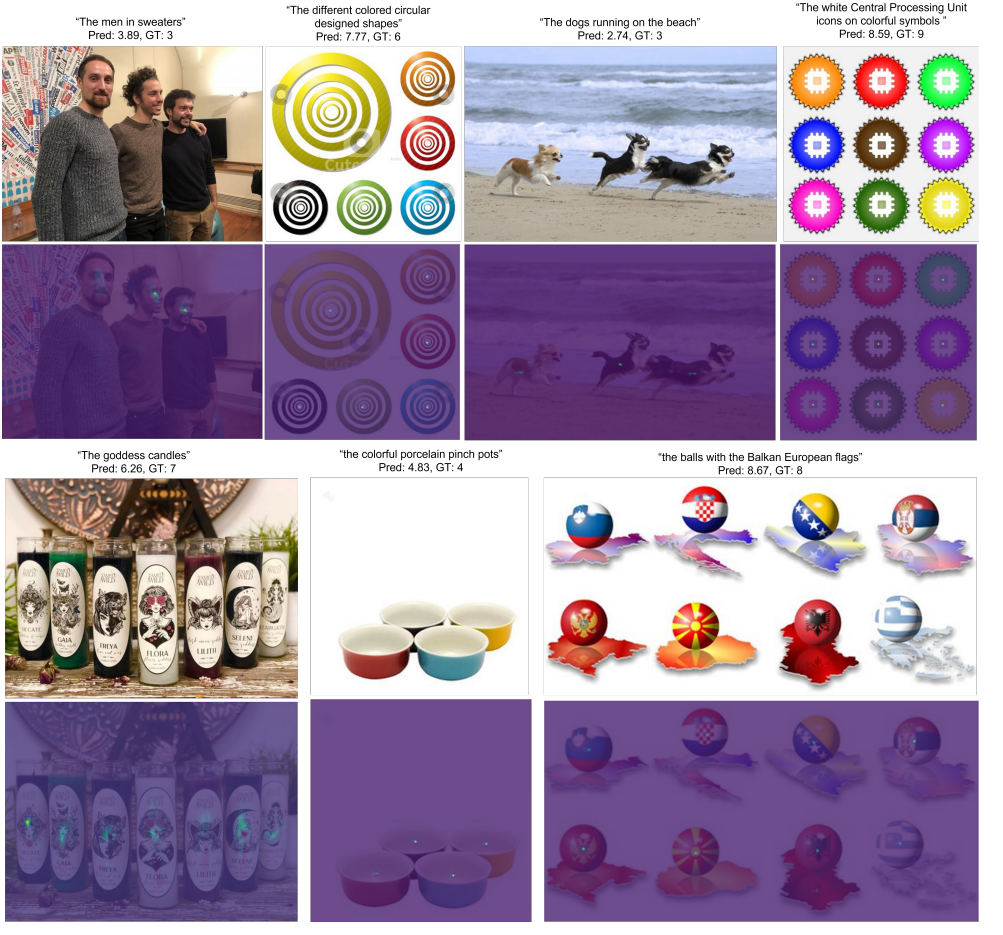

Figure 12: Density maps produced by CounTX when applied to the CountBench [11] subset. Many images in CountBench contain objects that only exist digitally such as the images of the icons and the balls with flags above. In contrast, FSC-147 only contains images of real objects. Despite this, CounTX is able to count the icons in the rightmost image in the top row. Notice how CounTX correctly places a dot on the large circle in the second image in the top row, even though it is much larger than the five other circles surrounding it.

7 Limitations

In this section, two weaknesses of CounTX will be discussed. CounTX struggles when an object is self-similar. Instead of counting each self-similar object as a whole, CounTX might double count by placing a dot on each similar component in the density map. For example, a typical pair of sunglasses is self-similar because it is composed of two similar lenses. As shown in Figure 13, instead of counting each pair of lenses as a whole object, CounTX might count each lens in the pair as an individual object. If visual exemplars were available, the final count could be calibrated by dividing the estimated count by the average sum of the

density map at each exemplar region. However, this is not currently possible with only text descriptions. Secondly, CounTX struggles to understand inter-object relationships. These weaknesses are illustrated and discussed in Figure 13.

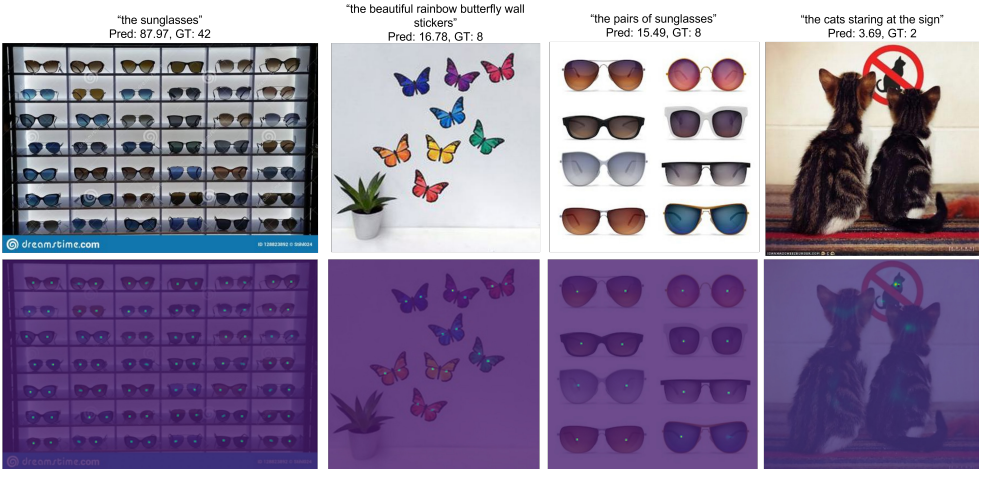

Figure 13: CounTX struggles to count self-similar objects. Instead of placing a dot on each pair of sunglasses in the density maps for the first image (from FSC-147 [12]) and the third image (from CountBench [1]), CounTX places a dot on each lens. This is why the estimated counts for these images are almost double the ground truth counts. Following this pattern, CounTX places a dot on each butterfly wing in the density map for the second image from CountBench above. This results in an estimated count that is almost twice the ground truth count. CounTX also struggles with inter-object relationships. This weakness surfaces when evaluating CounTX qualitatively on the CountBench subset as the text descriptions for images in the CountBench subset are more nuanced than the descriptions in FSC-147-D. In the rightmost image above from CountBench, CounTX incorrectly attempts to count all the cats in the image, real and illustrated, instead of just the real cats staring at the sign with an illustrated cat.

References

[1] Nikola Djukic, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation. *arXiv preprint arXiv:2211.08217*, 2022.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[3] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021.

[4] Ersin Kiliç and Serkan Ozturk. An accurate car counting in aerial images based on convolutional neural networks. In *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[5] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *eccv*, 2014.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, 2017.

[7] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. CounTR: Transformer-based generalised visual counting. In *Proc. BMVC*, 2022.

[8] Terrell Nathan Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *Proc. ECCV*, 2016.

[9] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[10] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021.

[12] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proc. CVPR*, 2021.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[14] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks Track*, 2022.

[15] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and ZHIGUO CAO. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *Proc. CVPR*, 2022.

[16] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *Proc. WACV*, 2023.

[17] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proc. CVPR*, 2022.